Advances in Cognitive Systems X (20XX) 1-6 Submitted X/20XX; published X/20XX

Real-Time Metacognition and Applications

Matthew D. Goldberg MDGOLD@CS.UMD.EDU
Donald Perlis PERLIS@CS.UMD.EDU
Computer Science Dept, University of Maryland, College Park, MD 20742 USA

Abstract

We explore several subtleties of metareasoning in real time and their formalization in active logic. These are then applied to new treatments of default reasoning and to reasoning about evolving beliefs of self and other agents. We also touch on generality of the approach taken here across agent types including humans.

1. Introduction

Metacognition – understood quite generally – refers to cognitions about cognitions, as in "I hate the fact that I am so forgetful." But commonly it is used to refer more specifically to knowledge and/or reasoning about an agent's knowledge and/or reasoning, whether one's own or of that of another. Metareasoning in particular is reasoning about (someone's) reasoning. For a broad general reference, see Cox & Raja (2011). This has been the focus of much attention in the AI literature, in the subarea of commonsense reasoning. It also illustrates a well-known strong methodological connection between AI and cognitive science.

Of the various so-called "aims" of AI, a major one is this: to foster deeper understanding of what constitutes a mind, by looking carefully at the sorts of processing that arise in trying to build systems (models) that can perform some of the sorts of things that minds appear to be able to do. Two key examples are learning and reasoning, and much of AI has been devoted to one or the other of these. With regard to reasoning in particular, AI – in the hands of people like McCarthy, Minsky, and Reiter among others – led to the discovery in the 1970s of the necessity and centrality of non-monotonic reasoning (NMR) in humans, something up until then almost entirely unexamined (and largely unsuspected). This showcases a major kind of contribution AI can make in cognitive science: revealing necessary types of processing – and suggesting specific mechanisms to perform them – that in the absence of computationally informed models would remain cloudy or even unknown.

With regard to NMR in particular, despite enormous progress on many fronts, the story is far from over. Standard models ignore at least two requirements for any realistic reasoning system (biological or artificial) deployed in the real world: (i) computational tractability, and (ii) the need for performing inferences that take account of time-passage as they are being performed.

Here is a quick example: *Is 79 a prime number? I don't immediately know, but after some thinking, I come to realize yes, it is. What about 779779779? That's a tougher one, and unless there's a quick trick (that eludes me at the moment) then I'll need lots of time to figure it out.* In

this example, the agent responding to the questions notes that it lacks an immediate answer, that it is engaged in thinking, that it has not yet finished, and then that it has come to a conclusion of sorts, including about an ongoing time-interval during which it may do more thinking. The fact that the topic is arithmetical is irrelevant, as the following example shows, drawn from but considerably extending Fahlman et al. (1981). We envision an agent (human or artificial) that knows (has in its KB) that cephalopods are a kind of mollusk, that mollusks have shells, and that cephalopods do not. In step 1 the agent is informed that X in particular is a mollusk. The agent proceeds to draw inferences as follows, where in step 3 additional information is given to the agent:

1. X is a mollusk.
2. So it has a shell.
3. X is a cephalopod.
4. So it doesn't have a shell.
5. My conclusion at step 2 is incorrect.
6. I should now regard an inference of the form above from 1 to 2 as non-monotonic, and withhold such a conclusion if I have information to the contrary.

While any one step here can be regarded as carried out (almost) instantaneously, the sequence reveals changes of mind that are themselves noted and reasoned about as that very sequence is unfolding in time. In addition, there is – however briefly – a moment (just after step 4) when the agent's knowledge is inconsistent: it has the beliefs that X does and doesn't have a shell, until it rectifies this in step 5. The trick in step 6 is to identify what constitutes "information to the contrary." Various formal treatments of NMR do this variously; but most suffer from violating both requirements mentioned above.

The active logic formalism was designed in part to address these shortcomings, and itself was an outgrowth of earlier work on so-called step-logics and an associated time-sensitive cognitive architecture for both memory and inference (Elgot-Drapkin et al., 1991). The present work digs more deeply into the inferential architecture; reveals complexities of time-sensitive NMR that were unsuspected beforehand (by us authors and by anyone else as far as we are aware); strongly hints that – somehow – our brains are also dealing with these complexities; and presents algorithms that address such complexities, as illustrated by several case studies. Among the latter are not only mollusk-type scenarios but also other-agent scenarios where one agent reasons in real time about the time-evolving beliefs of another agent.

Metacognition is very prominent in human cognition (see Nelson (1992)). A human (or artificial agent) benefits a great deal from being able to know and reason about what it (or another) is (or was earlier) thinking, and why. Consider reasoning about another agent, namely that *she is coming to realize X right now, because I am telling her exactly that at this very moment.* This illustrates a number of overlapping phenomena in cognitive neuroscience: (i) efference-copy, in which the brain retains a copy of efferent signals, that are then used to predict and assess outcomes of actions (as in *I know am speaking now*; see Brody et al. (2015)); (ii) the appearance-reality distinction (Flavell et al., 1983) as in *she thought X was false but X was actually true*; and the theory-theory of cognition (basically, in AI terms, that agents have KBs that can differ from one another, can change over time,

and can be wrong, as in *she is coming to realize X now*). Representing such types of knowledge and inference involves a new more rigorous treatment of subtle KRR (knowledge-representation and reasoning) control issues, especially with regard to nested forms of quotation, and has applications to time-situated NMR as well.

2. Time-Situated Metacognition

Automated systems often lack explicit knowledge of their own situatedness in the world – and often even lack knowledge that they are entities that have evolving beliefs and that can take actions. Such (typically missing) knowledge is a form of metacognition, and the ability to infer with and about it is a form of metareasoning. Alas, this lack is a severe limitation, arguably central to much of the brittleness of most automated systems: for instance, how can they correct their inevitable errors if they can't even recognize an error as arising from something they have done, or from something they believed incorrectly or did not yet know?

Metareasoning, like all reasoning, is an activity and thus takes place during time passage. Bringing this time-passage more fully into an agent's reasoning has been exploited regarding general commonsense reasoning in much work using the active logic formalism, with significant advantages including the ability to reason in the presence of contradictory beliefs, reasoning about approaching deadlines, and more; see for instance Purang (2001); Nirkhe (1995); Elgot-Drapkin et al. (1991); Josyula (2005).[1] But that same work has had mostly only informal ties to metareasoning. In particular, quotational mechanisms (affording powerful fine-grained representations of agent-beliefs adequate to support rich metareasoning) were given largely casual treatment, and as a result, many subtleties remained unexamined.

Work to incorporate elements of a "processual self" into active logic by Brody et al. (2014) was concerned with a much finer-grained notion of time, not from one step to another but rather *within* a step (viewed as a wff or sentence). This shares a broad similarity of aims to the present work, but without the issues of quotation or of NMR that are the main aspects we address here. Other recent work on metacognition (Anderson & Oates, 2007; Dannenhauer et al., 2018; Cox et al., 2021), either mostly does not represent time explicitly, or does not allow a general (quasi-)quotation mechanism or associated algorithm, which in particular hinders an agent's ability to reason about the evolving beliefs of another agent.

Here we report on new work, largely based on the recent PhD dissertation (Goldberg, 2022), that examines quotation and metareasoning much more closely in a real-time setting, with new applications to default reasoning (e.g., inferring a default rule from an incorrect monotonic rule) and also to reasoning about other agents' evolving beliefs within a natural-language scenario (inferring what agents do or don't know based upon what they have or haven't heard so far).

The rest of the paper is organized as follows: 1. how to represent quotation and quasi-quotation in active logic; 2. application to real-time treatment of defaults; 3. application to own-knowledge

1. Particularly salient here is active logic's avoidance of the "swamping" (or omniscience) problem, introduced in detail in Elgot-Drapkin (1988). That is, traditional logics have all logical consequences of their axioms as theorems, and thus are "swamped" by a (usually infinite) flood of mostly irrelevant (and nonsensical if the axioms are inconsistent) conclusions.

inferences; and 4. inferring other-agent evolving knowledge, with application to a problem due to E. Davis.

3. Quotation in Active Logic

For an agent to have explicit knowledge of – let alone reason about – its beliefs, it will need to represent a belief B in some fashion. If we consider an agent with a knowledge-base (KB), and B as an item within the KB, then that agent's knowing that its KB contains B would seem to require the KB to also contain something akin to InMyKB(B). This can lead to a powerful and fine-grained mechanism for representing knowledge of own-beliefs; but something is missing, at least if we are to employ a standard symbolic language such as that of first-order logic (FOL)[2]. For B, if it is a formula, cannot directly be an argument to a predicate. Quotation provides a way out: InMyKB("B"), B represents a formula, and where "B" is a "quotation-term" of which we will say more below.

We will assume the reader has some acquaintance with active logic,[3] which employs a variant of the language of FOL. Here is a very brief summary. Inferences are associated with step-numbers (typically positive integers) indicating the relative order in which they are inferred. Thus, there is an implicit inference-engine (e.g., resolution or some other similar inference-method) being applied to the current wffs ("beliefs") at a given time-step t to produce any new conclusions at time $t+1$. Thus the set of beliefs (KB) evolves, as in:

Step	Beliefs	Justification
1	Now(1), A, A $\rightarrow$ B, B $\rightarrow$ C, B $\rightarrow$ D	Initially given
2	Now(2), A, A $\rightarrow$ B, B $\rightarrow$ C, B $\rightarrow$ D, B	Modus ponens
3	Now(3), A, A $\rightarrow$ B, B $\rightarrow$ C, B $\rightarrow$ D, B, C, D	Modus ponens

Underlined wffs indicate ones newly inferred at that step.The step numbers themselves can enter into wffs, especially via the special Now predicate as illustrated. Notice that whereas wffs in general tend to be inherited from step t to $t+1$, Now(t) is not but rather is subject to a special "clock-rule" that specifies inferring Now(t + 1) from Now(t). A direct contradiction in the belief set (KB) yields another special (non-inherited) conclusion: if P and $\neg$P are present at step t, then neither is inherited but the wff Contra(t, "P", "$\neg$P") is inferred at $t+1$. An implementation of active logic is usually referred to as ALMA (with a version number), and the name Alma is informally used to refer to the underlying agent whose reasoning is performed in ALMA. ALMA2.0 is described in detail by Goldberg (2022) and more details are also found in the appendix: https://mclumd.github.io/Appendix/appendix.html.

Quotation terms, mentioned above, are new terms provided in ALMA2.0. Syntactically, a quotation term consists of a formula enclosed in parentheses, as in "W" where W is a wff. This is intended recursively, so that for instance "Believes(John, "W")" is itself a quotation term, and

2. There are modal alternatives, but these suffer from the aforementioned "swamping" – for instance, not allowing A $\rightarrow$ B and A to be in a KB without B also being there at the same time, which sidesteps any notion of inference being a process that an agent can represent as occurring over time.
3. See Perlis et al. (2017) for a quick introduction and Elgot-Drapkin & Perlis (1990) for a highly detailed technical account.

thus Believes(Jill, "Believes(John, "W")") is a wff as well.[4] Here we note a key point: all wffs written in the language of ALMA2.0 are intended to represent possible items in the (evolving) KB of the agent Alma. Thus inference of Believes(Jill, "Believes(John, "W")") indicates a belief by *Alma* about a belief of Jill. Likewise, inference of any W amounts to Alma coming to believe W. Thus inference in ALMA2.0 is inference *by* Alma. Goldberg (2022) provides a grammar that spells out the precise role of quotation terms in the ALMA2.0 language.

Using quotation terms, Alma can express its belief that it does or does not have a given wff W in its KB, as Know("W") or ¬Know("W"), which we had earlier but less clearly expressed in the form InMyKB(W). But how is Alma to infer that it does or does not know a particular formula that would appear in place of W? Here the key fact that Alma's (evolving) KB is always finite is crucial. It is then simply a lookup to determine whether or not a given wff W is a current element of the KB at timestep t, and the result of such a lookup can then be inserted as either Know("W") or ¬Know("W") at step $t + 1$. However, for efficiency purposes, it is useful to employ two special-purpose operators Pos_int and Neg_int, which when applied to a wff W perform the lookup and return true or false accordingly. In more detail, Pos_int("W") is used as an antecedent in a rule such as Pos_int("W") → P: if W is in the KB, then P. Pos_int is not inferred (not added to the KB) but rather it sanctions the inference of P. If it is desired for Alma to reflect that it in fact does know P, this can be accomplished via Pos_int("P") → Know("P"). Similarly for Neg_int.

Here we need to be more precise about syntax. In logic-programming, wffs are usually written in CNF, with (leading, universal) quantifiers being dropped (but implicitly assumed). Moreover, variables are capitalized and other symbols are not. Thus a FOL wff such as $\forall x[P(x)]$ becomes simply p(X). Context should make clear which notational convention is being used.

Alma can believe A, B, and A ∧ B → C at time t, and not yet believe C – until it has had time to make that inference. Yet a wff such as $\forall x, y[\texttt{Friends}(x, y)]$ should be co-believed with $\forall u, v[\texttt{Friends}(u, v)]$. Each "says" that everyone is everyone's friend; and so the two wffs should unify. And indeed they do. But what if there is quotation in the wffs, such as if Alma believes both believes(alice, "friends(X, Y)") and believes(alice, "friends(U, V)")?[5] The two beliefs attributed to Alice should unify – so long as ALMA's beliefs are not strictly about verbatim syntax. On the other hand, consider formulas believes(alice, "friends(X, Y)") and believes(alice, "friends(U, U)") – the latter of which expresses that Alice is considered to believe that everyone is their own friend, which is quite distinct from that everyone is everyone's friend. Unlike first-order variables outside of quotation, these two friends literals should *not* be unifiable. So unification for an agent's inferential processes must include special treatment of variables within quotes, which have some delicate considerations, due to the nature of quotation.

To determine more generally which terms ought to unify with a variable inside of quotation, it helps to again consider quantifiers made explicit in the formulas, which we recall means that universal quantifiers for quoted variables appear just within the nearest enclosing quotation marks.

4. In these examples, W should more precisely be replaced by an actual wff. Additionally, there is potential ambiguity in nested quote-marks; at a more precise (implemented) level, an intended pair "_" is replaced by quote(_), which eliminates ambiguity but makes reading cumbersome.

5. For which, if we make the quantifiers explicit, would be rendered as Believes(alice, "$\forall x, y[\texttt{Friends}(x, y)]$") and Believes(alice, "$\forall u, v[\texttt{Friends}(u, v)]$"). The universal quantifiers occur within the quotation terms, as Alma attributes *to Alice* beliefs that are universally quantified.

For two quoted formulas that are identical in structure except for a variable appearing in one formula and a ground term appearing in the second formula in the same places where this variable occurs, the second formula has at least one fewer quantifier within the quotation marks. If binding the variable in question to the ground term were allowed to occur, substituting occurrences of the variable for its binding would result in changing the formula structure by removing a quantifier, and hence yielding a more specific quoted formula. Since quoted formulas are in many ways treated as mentioned formulas that capture a greater degree of specificity, it is sensible that a quoted variable cannot unify with a ground term, preventing specialization when substituting. Similarly, a quoted variable should not unify with a non-ground function term containing other quoted variables, because these non-ground terms are also more specific quoted content than the universally quantified variable, regardless of whether the formulas they are embedded in may contain more universal quantifiers in total. Thus, a quoted variable may only unify with another quoted variable, and not another type of term.

The second of two principles characterizing quotation term unification reflects the intuitions of the example above, distinguishing the success in unifying "$\texttt{friends}(\texttt{X}, \texttt{Y}))$" with "$\texttt{friends}(\texttt{U}, \texttt{V}))$" from the failure to unify "$\texttt{friends}(\texttt{X}, \texttt{Y}))$" with "$\texttt{friends}(\texttt{U}, \texttt{U}))$". For two quotation terms to unify: 1. Their quoted formulas must have identical structures of operators, functions, literals, constants, and further-nested quotation terms (with the exception of quoted variables); and 2. For each quoted variable in one formula, all occurrences of this particular variable must correspond to exactly one variable in each corresponding location in the other formula.

Here an interesting distinction from other sorts of terms is evident: a variable within the formula of a quotation term may unify only with another variable, but neither a constant nor a more specialized pattern of variables. In contrast, a non-quoted variable does not behave as selectively for unification, and can unify with arbitrary constants or other terms that it subsumes.

3.1 Quasi-quotation

There is also a need for a more subtle form of quotation, called quasi-quotation. We introduce the use of quasi-quotation as a means to quantify into quotation, develop the desired unification behavior for quasi-quotation, and ultimately generalize this behavior to a full unification algorithm for quotation terms and quasi-quotation. This will be essential for representing an active-logic agent's reasoning about beliefs – both the agent's own, as well as those of others.[6]

Quotation terms as introduced thus far have a significant limitation, exemplified by a formula such as the following:

$$\texttt{bird}(\texttt{X}) \wedge \texttt{neg\_int}(\text{“}\neg\texttt{flies}(\texttt{X})\text{”}) \rightarrow \texttt{flies}(\texttt{X}) \tag{1}$$

We see that the same variable that appears within a quotation term cannot also appear outside of that quotation with the same meaning. Here, the X inside of neg_int's argument has its quantifier placed just inside the first quotation mark. Hence, quoted variables in their present form are locked inside of quotation, where these variables are mentioned only. This inaccessibility of variables inside quotation terms prevents their binding and substitution for anything other than other quoted

6. Quotation and quasiquotation are mechanisms allowing for subtle distinctions between use and mention. While we are not proposing that brains literally have quoting notations, it would seem that there must be some way in which use and mention are finely distinguished; otherwise metacognition seems impossible.

variables, by the definition of unification for quotation terms. Yet in other cases, it is desirable for particular variables within quotation terms to have their corresponding quantifiers appear beyond the quotation marks – the problem of quantifying into quotation. What is desired is a pattern of how certain variables, if indicated for this purpose, may quantify into quotation terms and interpolate within the quotation marks. Such a pattern would allow: 1) the same variable to appear both inside and outside of quotation, 2) these variables to make bindings with other sorts of terms, and 3) substitution into the context of quotation. These abilities clearly exceed what can be done with quotation terms that have no ability to quantify-in.

Quantifying-in provides a theoretical mechanism for variables to bind and substitute into the context of quotation terms. We use the term *quasi-quotation* for referring to a syntactic device allowing quantifying-in, whereby variables may be selectively indicated to "escape" the quotation terms that they are contained in. Quasi-quotation was originally introduced by Quine (1981), although Quine used the term quasi-quotation slightly differently, unlike our usage for a mark that essentially unquotes particular variables. We emphasize that this feature may be selectively applied to only some variables. Not all variables should escape from inside of a quotation term; instead it is often useful for universal quantification to remain inside of quotation marks. Quantifying-in via quasi-quotation enables targeted variables in a formula to escape their containing quotation term during inference and unification, so that these variables might be substituted for or bound. Quasi-quoting the second use of `X` in formula 1, within the quotation term, now gives a way to achieve the desired meaning: that all appearances of `X` are occurrences of the same variable.

We now introduce a formalism for quasi-quotation, as built by extending the existing quotation terms. Quantifying-in is delineated with a special quasi-quotation character (the quasi-quotation mark) appearing before a variable to modify it. The backtick (`` ` ``) is used for this mark, inspired by Lisp.[7] For instance, we solve the problem of quantifying into formula 1 by modifying its second conjunct into the following: `neg_int(“¬flies(` `X)”)`.[8]

A variable may be preceded by multiple quasi-quotation marks; the purpose of allowing multiple marks is to quantify into nested quotation terms in different ways.[9] For an instance of a variable located within N nested pairs of quotation marks, we refer to this variable instance as appearing in N *levels of quotation*. With multiple quasi-quotation marks, the means of quantifying-in is generalized so that each additional level of quotation added before a particular variable allows the variable to escape one further pair of quotation marks, from the innermost out. The implicit universal quantifiers for a quoted variable thus occur outward an additional level with each added quasi-quotation mark. An instance of a variable in N levels of quotation with N quasi-quotation marks is referred

7. Quasi-quotation in Lisp (and related dialects such as Scheme) is an alternative form of quotation in the language, which serves as a template whereby parameters may be inserted into the template by selectively unquoting expressions via another operator (Steele, 1990). Some other languages, such as Haskell, also support the feature due to inspiration from the Lisp family (Mainland, 2007). In the programming language domain, this feature is used for program-generating programs, to more easily embed domain-specific languages by specifying syntax and constructing fragments of programs (Bawden, 1999).
8. The reader is cautioned not to read the backtick here as a quote.
9. If instead of being able to repeatedly quasi-quote, a single quasi-quotation mark were to immediately allow a variable to fully escape from all nested levels of quotation, this would not have sufficient expressiveness for desired uses of quasi-quotation. It is often convenient for a variable within several nested quotation terms to be only partially quasi-quoted (i.e., not fully-escaping, and having an effective level of quotation greater than zero).

to as *fully-escaping*; any fully-escaping variable has a quantifier with scope over its entire formula. This scope is intuitive given that the effective level of quotation for a fully-escaping variable is zero. Hence, such a variable behaves quite similarly to a variable outside of any quotation.

Questions of which bindings would be permissible for quasi-quoted variables, and how substitution should occur in these contexts, must be answered to make use of quasi-quotation. Due to the richer structure of a quotation term with embedded quasi-quotation marks inside, there are more possibilities for how this might behave than in the more straightforward case of unifying a quotation term without quasi-quotation. Goldberg (2022) presents the full details of behavior. Here, we provide a condensed summary of results with regard to quasi-quotation unification and bindings.

A non-escaping variable can create a binding exclusively when matching with select cases of other non-escaping variables: when these variables have the same effective quotation level, and as long as there is consistency in variable matching. A binding is rejected when the term it is compared to is of any other type – a function, a constant, a quotation term, a fully-escaping variable, or a non-escaping variable violating the above conditions.

A fully-escaping variable might create a binding exclusively when matching with another fully-escaping variable, a constant, a function or a quotation term (as long as either of these types does not contain a variable with implicit quantifiers that escape outside the boundary of the term in question). Conversely, a binding is rejected when comparing to a non-escaping variable, for a function or quotation term with improper quantifier placement (according to the principle identified above).

Taken together, these cases give an exhaustive accounting for the behavior during unification involving a variable compared against any kind of term, whether success or failure results. We can also further provide an exhaustive accounting for unification behavior when comparing a pair of terms that are both non-variables: functions, constants, and quotation terms. With variables removed from consideration, unification is straightforward in that these terms simply successfully unify when their structures match, and otherwise fail to unify when there isn't a structural match. Hence without regard to variables, a constant unifies exclusively with an equal constant, a function unifies exclusively with another function of matching functor where each term recursively unifies, and a quotation term unifies exclusively with another quotation term containing matching predicates and where each predicate argument recursively unifies.

This behavior is made most explicit in the full algorithm presented by Goldberg (2022). In addition to the detailed algorithm, this also details bodies of examples for each principle of accepting or rejecting unification with quotation terms.

We also note that the form of unification augmented to fully account for quotation can be applied to Alma reasoning about its own beliefs, or lack of belief, at various times. A later section will give more detail, but for now a foreshadowing will be useful.

Basically, a self-query posed to Alma (intuitively, a question of the form "Do I believe `W`?") is a wff which Alma then tries to find in its KB. Here is a preliminary example: suppose ALMA has in its KB the general rule $\texttt{heard}(\texttt{X}) \wedge \texttt{pos\_int}(\texttt{X}) \rightarrow \texttt{agreement}(\texttt{X})$ – i.e., if I heard something and if it is also in my KB, then I am in agreement with what I heard – as well as also having in the KB $\texttt{heard}(\text{“}\texttt{on\_fire}(\texttt{site1})\text{”})$ and $\texttt{on\_fire}(\texttt{E})$. Recall that by the convention specified above, this latter wff has the meaning that everything is on fire. It would then seem appropriate for Alma to infer agreement with $\texttt{on\_fire}(\texttt{site1})$. Alas, $\texttt{on\_fire}(\texttt{site1})$ is not explicitly in the

KB. However, pos_int is implemented in such a way that they will evaluate to true as long as on_fire(site1) can unify with something in the KB; and similarly for neg_int (which returns false unless something can unify). In effect, Alma uses pos_int as a kind of self-query as to whether it believes the argument, before applying the above rule to conclude agreement. In a later section this idea is greatly expanded.

4. Real-time Default Reasoning

Why is default – or nonmonotonic – reasoning (NMR) a form of metareasoning? What does NMR have to do with metacognition or self-knowledge? Well, NMR always involves somehow basing conclusions on an assessment of what one does or doesn't already know. The "already" here is the key issue on which our treatment of default reasoning centers. At a rough gloss, NMR can be thought of as applying a rule like this: $\texttt{P} \wedge \neg\texttt{know}(\text{“}\neg\texttt{Q}\text{”}) \rightarrow \texttt{Q}$; that is, given P and the fact that I don't know ¬Q, I conclude Q. Thus, the idea is to express that when P is true, Q is reasonably assumed to be true as well, unless I happen to know otherwise. For instance, to use a modification of Minsky's famous example, $\texttt{bird(X)} \wedge \neg\texttt{know}(\text{“}\neg\texttt{flies(X)}\text{”}) \rightarrow \texttt{flies(X)}$: birds typically fly. In this particular example, we see the need for quasi-quotation so that x can work as a variable even while inside the quotation-term.

Approaches to NMR tend to treat $\neg\texttt{know}(\text{“}\neg\texttt{Q}\text{”})$ as a consistency test, namely indicating that ¬Q cannot be inferred from (is not a logical consequence of) whatever is already in the KB, showing Q to be consistent with the KB. While this is a powerful theoretical tool, it is impractical for an actual agent to use since consistency in general is uncomputable. McCarthy's work on circumscription (McCarthy, 1986) makes two major advances: a computable stand-in for consistency-testing (minimizing atypical predicates), and the conceptual simplification of substituting ¬abnormal for ¬know, as in this modification of the Minsky example: $\texttt{bird(X)} \wedge \neg\texttt{abnormal(aspect2(X))} \rightarrow \texttt{flies(X)}$, where aspect2(X) cryptically denotes something like the bird-flyingness of X. The minimization technique is then applied to the predicate abnormal.

Alas, while computable, circumscription involves considerable processing, making it unwieldy for real-time uses. The NMR alternative that we present here makes no attempt to characterize what might or might not be inferable from the current KB, but rather simply what is currently in the KB. Since this tends to change, then conclusions might flip-flop over time, instead of being a limiting case after all possible reasoning has been done (which is what leads to the uncomputable or unwieldy problems). Such flip-flopping may appear concerning; but we argue that this is desirable in a real-time reasoner such as Alma; as it comes to realize additional facts that show earlier conclusions false, it should indeed change its mind. The key point is that Alma (indeed any reasoner at all, human or artificial) cannot know something until in fact it actually knows it, and thus that changes of mind are essential and appropriate.[10]

Our approach then essentially replaces the above two wffs in the circumscription version with $\texttt{bird(X)} \wedge \neg\texttt{abnormal(X, bird, “}\neg\texttt{flies(X)}\text{”}) \rightarrow \texttt{flies(`X)}$, and $\texttt{neg\_int(“abnormal(`X, bird,}$

10. This of course is not to say that the reasoner should be oblivious to frequent changes of mind, or that it should not withhold judgment if, for example, it has reason to think a fuller answer might soon be forthcoming. But these are beyond the scope of the current effort.

“¬flies` `X)”) → ¬abnormal(X, bird, “¬flies(` X)”). These are easily generalized to fairly arbitrary predicates in place of bird and flies. There is one further technical issue: bird is being used as an argument to abnormal, but is supposed to somehow correspond to the predicate bird. There are various ways to achieve this; one highly general one is to use an overall “relation” predicate (rel) along with names to stand in for other predicates, as in:

$$\texttt{rel}(\texttt{Kind}, \texttt{Obj}) \wedge \texttt{neg\_int}(\text{“}\texttt{abnormal}(\texttt{`Obj}, \texttt{`Kind}, \text{“}\neg\texttt{rel}\texttt{``Prop}, \texttt{```Obj})\text{”})\text{”}$$
$$\rightarrow \texttt{rel}(\texttt{Prop}, \texttt{Obj}) \quad (2)$$

This expresses that if something (Obj, e.g., Tweety)) is of some type (Kind, e.g., a bird)) and it is not currently in the KB that Obj is an abnormal item of that type due to not having property Prop, then that Obj is concluded to have that property. Here Kind, Prop, and Obj are variables that can be replaced with, for example, the names bird, flies, and tweety. For future reference as used below, we can think of the Neg_int clause above as a “not-abnormal-not” insertion of typicality into what otherwise would have been a strict monotonic rule (such as that all birds fly).

With this machinery in hand, we can now show results of case studies of how Alma can make default inferences and adjust those conclusions as its KB acquires additional information – whether by new external observations coming into the KB or by its own reasoning. Moreover, at the same time Alma will convert an initially monotonic rule such as $\texttt{rel}(\texttt{P}, \texttt{X}) \rightarrow \texttt{rel}(\texttt{Q}, \texttt{X})$ into a default version: $\texttt{rel}(\texttt{P}, \texttt{X}) \wedge \texttt{neg\_int}(\text{“}\texttt{abnormal}(\texttt{`X}, \texttt{`P}, \text{“}\neg\texttt{rel}(\texttt{``Q}, \texttt{```X})\text{”})\text{”}) \rightarrow \texttt{rel}(\texttt{Q}, \texttt{X})$ if a counterexample (an object having property P but not property Q) is encountered. We will start with a famous example of nested defaults due to Fahlman but extended to deal with temporal evolution of knowledge, and then also consider a more complex case of overlapping defaults.

Fahlman et al. (1981) identified the need for a knowledge-based system to handle a series of nested default rules, where each more-specialized case contradicts more general knowledge. The specific example was provided with these rules: A mollusk typically is a shell-bearer; a cephalopod is a mollusk, but typically is not a shell-bearer; a nautilus is a cephalopod, but typically is a shell-bearer; and a naked nautilus is a nautilus but typically is not a shell-bearer.

Traditional nonmonotonic reasoning does not deal with actually undoing conclusions over an episode of reasoning. For example, if a particular animal is simply known to be a nautilus, then a traditional approach would conclude that it is a shell-bearer. But if the reasoning occurs over time, and if a creature is first learned to be a cephalopod, and later the belief is acquired that this creature is a nautilus as well, then the above rules lead to a contradiction regarding whether the creature is a shell-bearer or not. Ideally, this should be repaired by giving up the belief the creature is not a shell-bearer. Active logic’s handling of direct contradictions allows reasoning to proceed even if direct inconsistencies arise. Then, by using nesting to refer to the contradictory beliefs, specific reinstatements can be made to serve as contradiction-repairs (namely, a conclusion from a narrower category is preferred over a conflicting one from a broader category).

We present solutions to sample problems centered around reasoning with formulas for interacting defaults based initially on Fahlman’s problem of reasoning with mollusk categories in an ontology. Our solutions extend the scope of this latter problem in several ways: automatically revising non-default (i.e., absolute or monotonic) rules into defaults based on evidence; reinstating appropriate contradictands; solving a more complex case with overlapping (not fully nested) defaults. Our axioms also pervasively uses quotation terms and quasi-quotation terms.

Using the rel notation, one easily crafts formal axioms for non-default versions of the the Fahlman rules listed above; one such axiom is the following:

$$\texttt{rel}(\texttt{mollusk}, \texttt{X}) \rightarrow \texttt{rel}(\texttt{shell\_bearer}, \texttt{X}) \tag{3}$$

In addition, we postulate (but do not list here) several axioms characterizing the `is_a` relation as transitive and other similar elementary ones such as that observations (new facts coming in, e.g., about a nautilus being observed) are believed.

Such axioms then characterize a series of progressively narrower categories of mollusks (a cephalopod as a specialized kind of mollusk, etc.) and the status of these groups as being shell-bearing or not, but (for the moment) without admitting exceptions. Thus the existence of even one cephalopod will lead to a contradiction since, as a mollusk it will have a shell, and as cephalopod it will not. The point is for Alma to figure out that something has to "give" and that relaxing "mollusks have shells" to "mollusks typically have shells" does the trick, as in

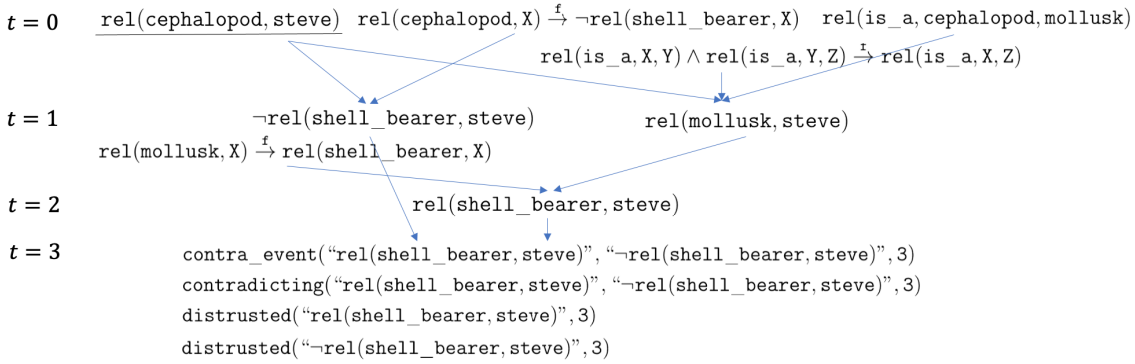
$$\texttt{rel}(\texttt{mollusk}, \texttt{X}) \wedge \texttt{neg\_int}(\text{“}\texttt{abnormal}(\texttt{`X}, \texttt{mollusk}, \text{“}\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{``X})\text{”})\text{”}) \rightarrow \texttt{rel}(\texttt{shell\_bearer}, \texttt{Ob}) \tag{4}$$

Full details get rather complex,[11] but a sketch is included in the discussion below.

The very first example we consider begins with information about a cephalopod arriving, in a form such as the following:

$$\texttt{rel}(\texttt{cephalopod}, \texttt{steve}) \tag{5}$$

For simplicity, we assume that this information arrived as an observation that the agent has made, and so the belief in Steve being a cephalopod is considered beyond reproach. After one timestep, $\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})$ is inferred; additionally, the belief that $\texttt{rel}(\texttt{mollusk}, \texttt{steve})$ is also inferred from the belief that a cephalopod is a mollusk, that is-a relations are transitive, and 5. After one more timestep, Steve also being a mollusk in combination with formula 3 produces that $\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})$. Thus the knowledge base now attributes both having and not having a shell to Steve, which is a direct inconsistency. The following figure expresses a trace of the relevant formulas over the timesteps in which this is obtained:



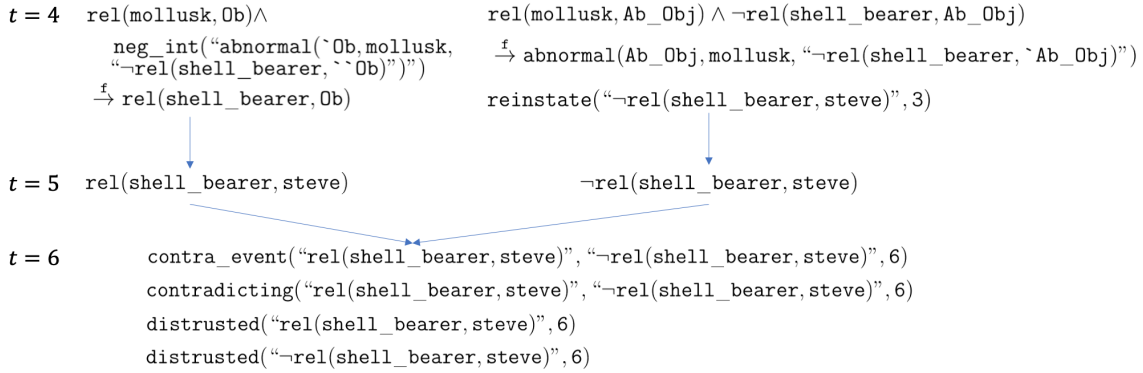
At this point, it is evident that one problem that led to the contradiction is that formula 3 is an absolute formula, rather than a default formula, and observing Steve has provided a counterexample.

11. See (Goldberg, 2022), where not only the axiomatization is given, but also runs of associated implemented code producing the stated results.

Thus, formula 3 should also be updated to amend it (using an "update" procedure) into a default formula. We achieve this effect through the following new axiom, which is lengthy due to being designed for generality:

$$\begin{aligned}
&\texttt{contradicting}(\text{“}\texttt{rel}(\texttt{`Pred}, \texttt{`Obj})\text{”}, \text{“}\neg\texttt{rel}(\texttt{`Pred}, \texttt{`Obj})\text{”}, \texttt{T}) \wedge \\
&\texttt{rel}(\texttt{Kind}, \texttt{Obj}) \wedge \texttt{rel}(\texttt{is_a}, \texttt{Kind_spec}, \texttt{Kind}) \wedge \\
&\texttt{parent}(\text{“}\texttt{rel}(\texttt{`Kind}, \texttt{O}) \rightarrow \texttt{rel}(\texttt{`Pred}, \texttt{O})\text{”}, \text{“}\texttt{rel}(\texttt{`Pred}, \texttt{`Obj})\text{”}, \texttt{T}) \wedge \\
&\texttt{parent}(\text{“}\texttt{rel}(\texttt{`Kind\_spec}, \texttt{`Obj})\text{”}, \text{“}\neg\texttt{rel}(\texttt{`Pred}, \texttt{`Obj})\text{”}, \texttt{T}) \\
&\quad \rightarrow \texttt{update}(\text{“}\texttt{rel}(\texttt{`Kind}, \texttt{Ob}) \rightarrow \texttt{rel}(\texttt{`Pred}, \texttt{Ob})\text{”}, \text{“}\texttt{rel}(\texttt{`Kind}, \texttt{Ob}) \\
&\qquad \wedge \texttt{neg_int}(\text{“}\texttt{abnormal}(\texttt{`Ob}, \texttt{` `Kind}, \text{“}\neg\texttt{rel}(\texttt{` ` `Pred}, \texttt{` `Ob})\text{”})\text{”}) \rightarrow \texttt{rel}(\texttt{`Pred}, \texttt{Ob})\text{”}) \\
&\quad \wedge (\texttt{rel}(\texttt{Kind}, \texttt{Ab_Obj}) \wedge \neg\texttt{rel}(\texttt{Pred}, \texttt{Ab_Obj}) \\
&\qquad \rightarrow \texttt{abnormal}(\texttt{Ab_Obj}, \texttt{Kind}, \text{“}\neg\texttt{rel}(\texttt{`Pred}, \texttt{`Ab_Obj})\text{”})) \qquad (6)
\end{aligned}$$

Here T is a time (or step) variable; and Obj, Ob, and O are variables ranging over possible indicating objects. An English summation of the premises is that the negative contradictand provides evidence that the narrower ontology category is a counterexample to this non-default parent formula, which states that for an object `Obj` in the wider category `Kind`, a positive `Pred` relation follows. The first conclusion is an update that inserts a "not-abnormal-not" typicality clause into the broader parent formula, making it into a default. Thus in the example, due to the existence of cephalopod Steve, the mollusk shell-bearing rule is updated to mollusks are typically shell-bearing. The following trace demonstrates these results (which also produce a secondary contradiction that must further be resolved):



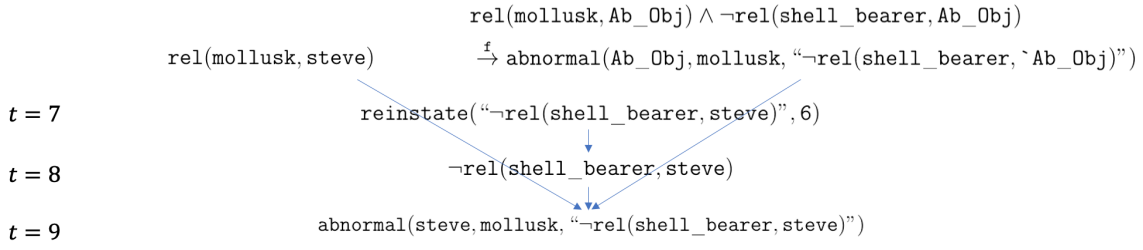
The second conclusion from formula 6 is an implication that gives the means by which other objects can be inferred to be abnormal, with respect to the predicates that have been used. For instance, in the example of Steve, this second conclusion is the following:

$$\begin{aligned}
&\texttt{rel}(\texttt{mollusk}, \texttt{Ab_Obj}) \wedge \neg\texttt{rel}(\texttt{shell_bearer}, \texttt{Ab_Obj}) \\
&\quad \rightarrow \texttt{abnormal}(\texttt{Ab_Obj}, \texttt{mollusk}, \text{“}\neg\texttt{rel}(\texttt{shell_bearer}, \texttt{`Ab_Obj})\text{”}) \qquad (7)
\end{aligned}$$

An object that is known to be a mollusk and which does not have a shell is thus concluded to be abnormal as a mollusk, with respect to being a shell-bearer, for this very reason of lacking one. As is the case with formula 4, this formula is specialized to particular predicates (i.e., `mollusk` and `shell_bearer`) in places where formula 6 had fully-escaping variables. Thus, formula 6 enables

the desired revision of a piece of ALMA ontology knowledge into a default rule (e.g., formula 4), and instantiates another belief that allows the concluding objects to be abnormal (e.g., formula 7).

Additional axioms (not given here) have been formulated to obtain further desired conclusions about Steve, e.g., that Steve is (now known to be) an abnormal mollusk. This involves the aforementioned notion of reinstating a formerly distrusted object-property contradictand belief (under special conditions). This can be summarized as reinstating the negative contradictand when it has been inferred from the narrowest ontology category out of the two contradictands. This encodes an intuition behind why Steve lacking a shell is given credence: a narrower ontology category might provide an exception to a rule that applies differently to a broader category in the hierarchy. The following trace provides the concluding inferences, following using formula 6 to update into defaults:



5. Own-Knowledge Inferences

ALMA has, in addition to `pos_int` and `neg_int`, three variants of each, for a total of eight in all, that together allow multiple forms of procedural lookup for a wff in the KB, including wffs that may have been in the KB earlier but were not inherited later on. The group of four `pos_int` predicates each evaluates to true when finding a matching unifiable formula in the knowledge base, although with some minor variations in which formulas satisfy each procedure. Similarly, the group of `neg_int` predicates each evaluates to true when failing to find such a formula, respectively. The variants restrict unification to particular cases, allowing more fine-grained control when needed. Using these in various combinations, we have developed a set of axioms[12] that endow ALMA with abilities for basic question-answering regarding whether a query formula `X` is believed by the Alma agent. Question-answering may be best thought of not as answering a question posed by another agent (human or otherwise), but as self-questioning in which the Alma agent is (agentively) assessing what it does or doesn't know – although precisely when and why an agent asks itself about particular beliefs is beyond the scope of this work.

ALMA uses the `pos_int_past` and `neg_int_past` procedures as part of question-answering if past beliefs are relevant to the query – alongside other procedures including `acquired`, `pos_int`, and `neg_int`. We note that some of the question-answering axioms rely on the convention of `rel` as a predicate symbol, with its first argument as a constant standing for a particular predicate symbol.

The query predicate, `query_belief`, is a binary predicate in which the first argument is the query formula, Q, and the second is the timestep for which the query is asked. Each query is

12. Alas, too lengthy for inclusion here; see Goldberg (2022) for details.

ultimately answered with a formula of the schema $\texttt{answer}(\text{“}Q\text{”}, A, \texttt{reason}(\text{“}R\text{”}))$, in which A will be one of the constants `yes`, `no`, or `unsure`, and R provides a formula that justifies the answer. Whenever possible, a current belief is provided for this reason R.

Here is an example of the above in use, based on adding question-answering queries to a default reasoning example, in which formulas believed regarding a cephalopod change over time. In this example, a query is raised each timestep, where only the timestep argument changes; for example:

$$\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 3) \tag{8}$$

As the beliefs regarding Steve change over time, we see the query answers change as well.

The first query in effect occurs at timestep 3, when there has occurred a contradiction between $\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})$ and $\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})$. In this case, due to the contradiction, the query is ultimately answered as follows:

$$\begin{aligned}&\texttt{answer}(\text{“}\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 3)\text{”}, \texttt{no},\\ &\quad\texttt{reason}(\text{“}\texttt{contradicting}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”},\\ &\quad\text{“}\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 3)\text{”}))\end{aligned} \tag{9}$$

$$\begin{aligned}&\texttt{answer}(\text{“}\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 3)\text{”}, \texttt{no},\\ &\quad\texttt{reason}(\text{“}\texttt{distrusted}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 3)\text{”}))\end{aligned} \tag{10}$$

The answers to the query remain the above pair until a later timestep, at which point both $\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})$ and $\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})$ have been re-derived. As a result, the queries have been answered in a contradictory manner:

$$\begin{aligned}&\texttt{answer}(\text{“}\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 5)\text{”}, \texttt{yes},\\ &\quad\texttt{reason}(\text{“}\texttt{acquired}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 5)\text{”}))\end{aligned} \tag{11}$$

$$\begin{aligned}&\texttt{answer}(\text{“}\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 5)\text{”}, \texttt{no},\\ &\quad\texttt{reason}(\text{“}\texttt{acquired}(\text{“}\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 5)\text{”}))\end{aligned} \tag{12}$$

However, when the contradiction between the `shell_bearer` pair is also detected at timestep 6, the above pair of query answers are also distrusted.

At timestep 8, the negative contradictand has been reinstated, and thus at timestep 9 the query answer reflects this fact, that the positive contradictand remains distrusted as a belief, while its negation is now believed:

$$\begin{aligned}&\texttt{answer}(\text{“}\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 8)\text{”}, \texttt{no},\\ &\quad\texttt{reason}(\text{“}\texttt{acquired}(\text{“}\neg\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 8)\text{”}))\end{aligned} \tag{13}$$

$$\begin{aligned}&\texttt{answer}(\text{“}\texttt{query\_belief}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 8)\text{”}, \texttt{no},\\ &\quad\texttt{reason}(\text{“}\texttt{distrusted}(\text{“}\texttt{rel}(\texttt{shell\_bearer}, \texttt{steve})\text{”}, 6)\text{”}))\end{aligned} \tag{14}$$

At this point, the cephalopod knowledge has reached a steady state, and ALMA idles.

6. Inferring Other-agent Knowledge

Endowing Alma with reasoning about the evolving beliefs of another agent has added complexities, involving several kinds of knowledge as well as reasoning in real time about presumed effects of actions (communication) on others' internal states. We will examine this within a single scenario,

proposed by Davis (2001); but our approach is intended to be fairly broad and should apply to a much wider range of applications.

A key to our approach is Alma having – for each agent that it knows of – a special partition in Alma's KB to hold beliefs that Alma attributes to that agent, and to which ALMA will apply its inference mechanisms to infer conclusions within that partition. Thus if Alma's "view" of Bob is that Bob believes `P` and `P` $\rightarrow$ `Q` at time t, then that same partition will also believe `Q` itself at time $t+1$. This nicely obviates the need for Alma to explicitly reason about Bob's individual inferences.

The Surprise Birthday Present (SBP) problem was devised by Davis (2001). It has various aspects; one such – given emphasis by Davis, and later by Morgenstern (2005) – is that of characterizing what it is for something to be a surprise (to some other agent) in the future. Another is how an agent can infer, based on such a characterization, that something will be a surprise to another agent. The SBP is roughly as follows (Davis gives several versions): Alma and Bob want to give Carol a birthday present, but do not want her to know what it is in advance. Alma and Bob communicate about an idea for a possible present. If Carol is not present during this communication, how can Alma infer 1.) that Carol will be surprised by the specific present (i.e., will not know what it is beforehand), 2.) that Bob will not be surprised, 3.) that Bob will also infer that Carol will be surprised? And conversely, if Carol is present, how can Alma infer that she will not be surprised, that Bob will also infer this about Carol, etc?

This may seem almost trivial, but the knowledge-representation details are not so by any means. We consider that Alma and Bob want to surprise Carol, and the issue is how Alma can represent and reason about what Alma believes are Bob's and Carol's evolving beliefs, as Alma and Bob have made this decision. For the part of Alma modeling Carol, if Alma believes Carol is nearby to be able to overhear what is told to Bob, then intuitively Carol ought to be able to determine that the gift is coming — and by expecting the gift, Carol fails to be surprised. Conversely, if Alma does not believe Carol to have had an opportunity to overhear, then Carol will not be modeled as ultimately unsurprised. Additionally, if Alma considers Bob to have much the same knowledge of the situation, then Alma ought to also model Bob reaching the same conclusion about whether Carol fails to be surprised. We are thus considering a need for Alma to be in possession of certain knowledge relevant to this commonsense situation — such as when an agent might hear information, when an agent is not surprised due to an expectation of a gift, and so on. But additionally, Alma should also be able to attribute to Bob and Carol the same knowledge of the situation.

Our short presentation here will necessarily be informal, but the ideas have been given precise formulation and implementation. Among other things, Alma will need a concept of common-knowledge, to be able to assume another agent has a certain minimum base of knowledge shared with Alma. We confine our discussion to the case of Alma, Bob, and Carol as in the SBP.

Alma has the belief that if a wff is common knowledge then: 1. it is true, and 2. other agents will also believe that it is common knowledge and that it is true. This avoids a need for Alma's axiom set to repeat distinct formulas formulas attributing knowledge of the situation to Bob or Carol. Alma simply must be aware of the existence of its fellow agents, and believe that scenario axioms are common knowledge, and the beliefs will be appropriately passed to the agent partitions.

The following axioms (mostly rendered in English here) constitute the initial beliefs of an agent for the specific problem of giving a surprise birthday present. These axioms model how agents that

heard an utterance may attribute hearing (or not hearing) to other agents: 1. Any agent near the speaker (including the speaker themself) will have heard the utterance. 2. An agent not near the speaker will not have heard the utterance by the speaker. Alongside the formulas for attribution of hearing (or not hearing), there are formulas expressing their consequences.

We assume that agents are credulous and believe what they hear in our scenarios, without considering reasoning related to falsehood (and thus, the giving of a surprise birthday gift does not depend on deceiving Carol). An agent might believe another heard an utterance for several reasons. When this is the case, the conclusion is that other agents would believe in what the present agent expects them to have heard. Similarly, if another agent is considered by the present agent to not have heard an utterance, this other agent is modeled as lacking a belief in the utterance.[13]

Next are the critical two axioms for determining when there will be a failure to surprise Carol with the gift. First, there is the common knowledge that an agent aware of a decision for giving a gift will expect that the recipient will receive it. Here, expectation is a shorthand for encoding that, as a default, its argument is likely to happen in the future, although condensed for readability

$$\texttt{common\_knowledge}(\text{“}\texttt{decision}(\texttt{Giver}, \texttt{give}(\texttt{Gift}, \texttt{Agent}))$$
$$\rightarrow \texttt{expectation}(\texttt{receive}(\texttt{Agent}, \texttt{Gift}))\text{”}) \tag{15}$$

The expectation that a certain agent will receive a gift might be believed by any agent. When the expectation that someone will receive the gift is held by the ultimate recipient of the gift, who moreover has heard about the decision to give it to them, this recipient will not be surprised when they receive their gift:

$$\texttt{common\_knowledge}(\text{“}\texttt{expectation}(\texttt{receive}(\texttt{Recipient}, \texttt{Gift}))\wedge$$
$$\texttt{heard}(\texttt{Recipient}, \text{“}\texttt{decision}(\texttt{`Giver}, \texttt{give}(\texttt{`Gift}, \texttt{`Agent}))\text{”}, \texttt{Speaker})$$
$$\rightarrow \neg\texttt{future\_surprise}(\texttt{Recipient}, \texttt{gift}(\texttt{Gift}))\text{”}) \tag{16}$$

Inference of the conclusion of axiom 16 by an agent model of Carol thus naturally represents a failure of the desired result by Alice and Bob.

Lastly, there is a small collection of axioms necessary for the scenarios to function: It is common knowledge that the three agents exist, that being near is a symmetric relation, and also that each agent believes and is identified by their own name. We suppress details of these.

Several versions of SBP have been examined, but all begin with the following formula of Alma telling an utterance to Bob:

$$\texttt{tell}(\texttt{alma}, \text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”}, \texttt{bob}) \tag{17}$$

In practice when these scenarios are executed by ALMA, due to the axioms for inferring the truth of common knowledge, for each axiom of the form $\texttt{common\_knowledge}(\text{“}X\text{”})$ (which would be believed by Alma itself and within a modeled agent's KB), there is a several step derivation of $\texttt{true}(\text{“}X\text{”})$ following, and then X alone as a belief. For simplicity, we refer to X as a formula extracted from common knowledge, in relation to $\texttt{common\_knowledge}(\text{“}X\text{”})$.

As with the earlier default reasoning examples, the subsequent discussion is at a high level, without full detail of inferences. Here we present only one such example; full traces for ALMA execution can be found in the available source code. We consider when all three agents are mutually

13. This obviously is an overstatement, since many un-uttered statements will be believed. For the present scenario, all that matters is that agents do or don't hear about the intended present and thus do or don't know about it.

known to be nearby. This intuitively should ultimately produce a failure to surprise Carol, since she will be able to overhear the utterance and should make inferences based upon what is heard.

First, we break down how reasoning proceeds for Alma expecting that Carol should be unsurprised. From a belief that a speaker hears their utterances, Alma infers the following:

$$\texttt{heard}(\texttt{alma}, \text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”}, \texttt{alma}) \tag{18}$$

From formula 18, the axiom $\texttt{agentname}(\texttt{alma})$ of an agent accessing its own name, and beliefs extracted from common knowledge, Alma infers $\texttt{true}(\text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”})$, and subsequently believes its nested formula:

$$\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol})) \tag{19}$$

Formula 19 satisfies the premise of 15, producing this conclusion indicating that Alma expects Carol to receive the gift: $\texttt{expectation}(\texttt{receive}(\texttt{carol}, \texttt{cake}))$.

In a second derivation thread, Alma infers that Carol has also heard the utterance:

$$\texttt{heard}(\texttt{carol}, \text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”}, \texttt{alma}) \tag{20}$$

Yet, due to the timing of inference and the need to unpack formulas from `common_knowledge`, at the timestep when 20 is inferred, one timestep earlier its negation has been inferred based on a default rule about what hasn't been heard and a lack of knowledge at that timestep that Carol was near. A contradiction thus occurs between whether Carol has heard or not heard, which is subsequently resolved. Following this reinstatement, between the reinstated formula 20, as well as the expectation regarding Carol and 16 extracted from common knowledge, Alma infers that Carol won't be surprised by receiving the gift:

$$\neg\texttt{future\_surprise}(\texttt{carol}, \texttt{gift}(\texttt{cake})) \tag{21}$$

We next address how the KB partition modeling Carol (which is itself a model that is part of Alma's KB) is employed for the first scenario, and trace how reasoning proceeds in Alma's KB partition for the modeling of the beliefs of Carol. This process begins with an inference from Alma rather than the model of Carol, leading to Carol modeled as believing that she heard Alma's utterance:

$$\texttt{believes}(\texttt{carol}, \text{“}\texttt{heard}(\texttt{carol}, \text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”}, \texttt{alma})\text{”}) \tag{22}$$

Per ALMA mechanisms for agent modeling, this belief is synchronized into the partition modeling the beliefs of Carol, as $\texttt{heard}(\texttt{carol}, \text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”}, \texttt{alma})$. Additionally, Alma models Carol as knowing her own name, which leads to the following:

$$\texttt{believes}(\texttt{carol}, \text{“}\texttt{agentname}(\texttt{carol})\text{”}) \tag{23}$$

This is also synchronized into the partitioned region for the model of Carol, as $\texttt{agentname}(\texttt{carol})$. And lastly, for the final Alma inference that affects the model of Carol for the scenario, Alma infers that Carol will be able to extract common knowledge:

$$\texttt{believes}(\texttt{carol}, \text{“}\texttt{common\_knowledge}(\texttt{X}) \rightarrow \texttt{true}(\texttt{X})\text{”}) \tag{24}$$

The formula $\texttt{common\_knowledge}(\texttt{X}) \rightarrow \texttt{true}(\texttt{X})$ is inserted into the model of Carol, and from here this model can extract formulas from common knowledge in the same manner as is used by Alma itself to withdraw the formulas and believe them.

The inferences themselves from Carol's perspective continue in the nested KB for her beliefs. From the combination of her belief in hearing the utterance, her belief in her name, and a belief in the truth of utterances as extracted from common knowledge, the modeled Carol infers that $\texttt{true}(\text{“}\texttt{decision}(\texttt{alma}, \texttt{give}(\texttt{cake}, \texttt{carol}))\text{”})$, and subsequently infers the formula

decision(alma, give(cake, carol)). In contrast to 19, this formula is specific to Carol, and hence Alma would instantiate believes(carol, "decision(alma, give(cake, carol))"). Carol's belief in her copy of the commonsense implication 15, alongside knowledge of the decision, produces an inference that she expects to receive the gift:

$$\texttt{expectation}(\texttt{receive}(\texttt{carol}, \texttt{cake})) \tag{25}$$

Unlike the perspective for Alma, there is no contradiction reached regarding heard literals in the model of Carol. Therefore, from formula 25, Carol's equivalent of 22, and Carol's belief of 16 extracted from common knowledge, Carol also infers that receiving the gift won't surprise her:

$$\neg\texttt{future\_surprise}(\texttt{carol}, \texttt{gift}(\texttt{cake})) \tag{26}$$

Hence, in addition to Alma ultimately reasoning that Carol should not be surprised when she ultimately receives the gift, Alma also models that *Carol herself* can reason out this same fact as a consequence of being nearby and able to overhear Alma's utterance about the gift. Formula 26 will instantiate into the core KB the following formula, which we contrast with formula 21 to highlight the difference between the result from modeling Carol with the reasoning without her model:

$$\texttt{believes}(\texttt{carol}, \text{“}\neg\texttt{future\_surprise}(\texttt{carol}, \texttt{gift}(\texttt{cake}))\text{”}) \tag{27}$$

In fact, in this contrast we see a difference that resembles the distinction we have drawn between internal and external accounts of reasoning; formula 26 is an internal representation for the model simulating Carol's reasoning processes about a failure of surprise, while formula 21 is an external representation outside of the model of Carol.

In the same running scenario, Bob, as another agent near Alma when the utterance is made, is also modeled by Alma as hearing the utterance, and then modeled as believing the set of common knowledge formulas following their synchronization into Bob's partition. Thus, the inference process also leads to Bob's model expecting Carol to be unsurprised in the future (i.e., Bob's model producing its own final inference of the form of formula 26), along essentially the same lines.

More interesting is the fact that, due to beliefs of Bob's model including a copy of formulas for common knowledge, and the common knowledge of the other agents and their positioning near each other, the model of Bob will also further be able to attribute these common knowledge beliefs to a more deeply-nested partition for Carol. Thus, the inference for Bob's model having its own inner model of Carol proceeds with the same inference process as was used by Alma modeling Carol. After sufficient timesteps, Bob's model of Carol infers formula 26, Bob's model obtains the equivalent of 27, and the core KB obtains the following:

$$\texttt{believes}(\texttt{bob}, \text{“}\texttt{believes}(\texttt{carol}, \text{“}\neg\texttt{future\_surprise}(\texttt{carol}, \texttt{gift}(\texttt{cake}))\text{”})\text{”}) \tag{28}$$

In ALMA, each agent model's reasoning can progress simultaneously. However, since there is a longer temporal chain of inference for inner agent models, the core KB will obtain a formula such as 21 at timestep 8, before 27 at timestep 12 and formula 28 later still at timestep 14. Alma will thus acquire the belief Carol modeled as failing to be surprised before the model of Bob reasons fully to this conclusion. It is an opportunity for future work in which Alma can anticipate Bob's model making the same conclusions. More broadly, patterns of reasoning of this kind illustrate how the ALMA reasoner has the capability for reasoning by one agent over time as its own beliefs, as well as those of other agents, are undergoing inferential changes, and having those changes reflected upon.

A second scenario we have studied involves common knowledge that Alma is near Bob but no other knowledge of nearness. Then Alma and Bob both infer by default that Carol is not nearby,

and thus has not overheard their plans, and so will be surprised. Another scenario involves common knowledge that Alma is near Bob, but where only Alma and Carol have shared (not common to Bob) knowledge that Carol is also nearby. Since Alma and Carol know each other to be near, Alma is able to model Carol such that Alma infers Carol will not be surprised. Similarly, (Alma's model of) Carol is able to conclude that Carol knows she will not be surprised. However, Bob lacks the ability to reach this conclusion.

7. Conclusions

We have described the importance of quotation and quasi-quotation in metareasoning and presented methods to represent them suitably for inference in active logic. This includes a formal concept of variables escaping suitable levels of quotation. This was then applied to produce new real-time methods of default reasoning, including the updating of non-default rules into defaults when exceptions arise, with application to nested defaults. Further application was presented for reasoning about own and others' evolving beliefs, and illustrated in the Surprise Birthday Present problem.

For future work, we note that the default-reasoning work for nested defaults does not extend to overlapping (non-nested) defaults. We anticipate that ideas in the causation-based solutions to the Yale Shooting Problem (Hanks & McDermott, 1987) will provide useful gains in a number of cases.

Broad aspects of the KRR subtleties that were addressed here in nested and time-varying defaults, as well as in other-agent cognition, would seem to be essential to commonsense reasoning. This would be not only with regard to active logic, but in human-level artificial agents generally, and likely to humans as well. While specific details may well vary across agent types, the need to represent and keep track of rapidly changing past, present, and (anticipated) future beliefs and inferences seems impossible to circumvent, except in severely constrained scenarios.

References

Anderson, M. L., & Oates, T. (2007). A review of recent research in metareasoning and metalearning. *AI Magazine*, *28*, 12–12.

Bawden, A. (1999). Quasiquotation in Lisp. *Partial Evaluation and Semantic-Based Program Manipulation* (pp. 4–12). Citeseer.

Bibel, W. (2008). Transition logic revisited. *Logic Journal of IGPL*, *16*, 317–334.

Brody, J., Cox, M. T., & Perlis, D. (2014). Incorporating elements of a processual self into active logic. *2014 AAAI Spring Symposium Series*.

Brody, J., Perlis, D., & Shamwell, J. (2015). Who's Talking?–Efference copy and a robot's sense of agency. *2015 AAAI Fall Symposium Series*.

Cox, M. T., Mohammad, Z., Kondrakunta, S., Gogineni, V. R., Dannenhauer, D., & Larue, O. (2021). Computational metacognition. *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*. Cognitive Systems Foundation.

Cox, M. T., & Raja, A. (Eds.). (2011). *Metareasoning: Thinking about thinking*. MIT Press.

Dannenhauer, D., Cox, M. T., & Munoz-Avila, H. (2018). Declarative metacognitive expectations for high-level cognition. *Advances in Cognitive Systems*, *6*, 231–250.

Davis, E. (2001). The surprise birthday present problem. `commonsensereasoning.org/problem_page.html#surprisebirthday`. Accessed: 2021-8-25.

Elgot-Drapkin, J., Miller, M., & Perlis, D. (1991). Memory, reason and time: The Step-Logic approach. In R. C. Cummins (Ed.), *Philosophy and ai*, 79–103. Cambridge: MIT Press.

Elgot-Drapkin, J. J. (1988). *Step-logic: Reasoning situated in time*. Doctoral dissertation, University of Maryland at College Park, College Park, MD, USA.

Elgot-Drapkin, J. J., & Perlis, D. (1990). Reasoning situated in time i: basic concepts. *J. Exp. Theor. Artif. Intell.*, *2*, 75–98.

Fahlman, S. E., Touretzky, D. S., & Van Roggen, W. (1981). Cancellation in a parallel semantic network. *IJCAI* (pp. 257–263).

Flavell, J. H., Flavell, E. R., & Green, F. L. (1983). Development of the appearance-reality distinction. *Cognitive psychology*, *15*, 95–120.

Goldberg, M. D. (2022). *Time-situated metacognitive agency and other aspects of commonsense reasoning*. Doctoral dissertation, University of Maryland, College Park.

Hanks, S., & McDermott, D. (1987). Nonmonotonic logic and temporal projection. *Artificial intelligence*, *33*, 379–412.

Josyula, D. P. (2005). *A unified theory of acting and agency for a universal interfacing agent*. Doctoral dissertation, University of Maryland, College Park.

Mainland, G. (2007). Why it's nice to be quoted: quasiquoting for Haskell. *Proceedings of the ACM SIGPLAN workshop on Haskell workshop* (pp. 73–82). New York, NY, USA.

McCarthy, J. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artif. Intell.*, *28*, 89–116.

Morgenstern, L. (2005). A first-order axiomatization of the surprise birthday present problem: Preliminary report. *Seventh Int Symp on Logical Formalizations of Commonsense Reasoning*.

Nelson, T. O. (1992). *Metacognition: Core readings.*. Allyn & Bacon.

Nirkhe, M. (1995). *Time-situated reasoning within tight deadlines and realistic space and computation bounds*. Doctoral dissertation, University of Maryland, College Park, MD.

Perlis, D., Brody, J., Kraus, S., & Miller, M. (2017). The internal reasoning of robots. *Thirteenth International Symposium on Commonsense Reasoning*.

Purang, K. (2001). *Systems that detect and repair their own mistakes*. Doctoral dissertation, University of Maryland, College Park.

Quine, W. (1981). *Mathematical logic, revised edition*. Harvard University Press.

Steele, G. (1990). *Common LISP: The language*. Elsevier.